浙江大学
ZHEJIANG UNIVERSITY

ARC Topics

Wenzhi Chen, Zhongyong Lu
lzy6032@zju.edu.cn

Topic 1: Gem5 SE mode investigation

- Gem5 can simulate a complete system with devices and an operating system in full system mode (FS mode), or user space only programs where system services are provided directly by the simulator in syscall emulation mode (SE mode).
- **Task: Analyze the internal principle of Gem5 SE mode and try to implement some unfinished system calls. Target ISA can be Alpha, ARM and X86.**

Topic 2: Gem5 CPU model 1

- Gem5 supports AtomicSimpleCPU, TimingSimpleCPU, InOrderCPU and O3CPU. AtomicSImpleCPU and TimingSimple are simple CPUs. They can model single threaded 1 CPI machine. They are used to fast-forward simulation, warm up caches where detailed CPU modeling is not required. InOrderCPU and O3CPU belong to detailed CPU models. They model the timing for each pipeline stage. They are slower than simple CPUs but much more accurate.
- **Task: Analyze AtomicSimpleCPU and TimingSimpleCPU model of Gem5**

Topic 3: Gem5 CPU model 2

- Task: As in topic 2, analyze InOrderCPU model of gem5.

Topic 4: Gem5 CPU model 3

- Task: As in topic 2, analyze O3CPU model of gem5.

Topic 5: Memory system - Classic

- Gem5 supports two memory system models: classic memory system and ruby memory system. The Classic memory system model provides gem5 a fast and easily configurable memory system at the cost of accuracy and flexibility. Cache coherence is maintained using an abstract snooping protocol where probes across the memory hierarchy functionally occur instantaneously. Ruby memory system is more powerful.
- **Task: Analyze the Classic memory system model of gem5**

Gem5 Ruby memory system

- Ruby implements a detailed simulation model for the memory subsystem. It models inclusive/exclusive cache hierarchies with various replacement policies, coherence protocol implementations, interconnection networks, DMA and memory controllers, various sequencers that initiate memory requests and handle responses. The models are modular, flexible and highly configurable.
- Ruby memory system supports various flexible cache coherence protocols using SLICC, a domain specific language.

Topic 6: MESI_CMP_directory cache coherence

- MESI_CMP_directory: single chip, 2-level caches, strictly-inclusive hierarchy.
- Task: Analyze MESI_CMP_directory cache coherence protocol of gem5 Ruby memory system.

Topic 7: MOESI_CMP_directory cache coherence

- MOESI_CMP_directory: multiple chips, 2-level caches, non-inclusive (neither strictly inclusive nor exclusive) hierarchy.
- Task: Analyze MOESI_CMP_directory cache coherence protocol of gem5 Ruby memory system.

Topic 8: MOESI_CMP_token cache coherence

- Task: Analyze MOESI_CMP_token cache coherence protocol of gem5 Ruby memory system.

Topic 9: MOESI_hammer cache coherence

- MOESI_hammer: single chip, 2-level private caches, strictly-exclusive hierarchy.
- Task: Analyze MOESI_hammer cache coherence protocol of gem5 Ruby memory system.

Topic 10: Implement a new cache policy

- The Cache Memory models a set-associative cache structure with parameterizable size, associativity, replacement policy. L1, L2, L3 caches (if exists) in the system are instances of Cache Memory. The Cache Replacement policies are kept modular from the Cache Memory, so that different instances of Cache Memory can use different replacement policies of their choice. Currently two replacement polices -- LRU and Pseudo-LRU -- are distributed with the release. In this experiment, you should understand how a cache replacement policy operates and implement a new advanced replacement policy such as DIP, DRRIP.
- **Task: Implement a new cache replacement policy such as DIP and DRRIP.**

Gem5 network

- There are two network models in Ruby, Simple and Garnet. The default model is the simple network. The simple network models hop-by-hop network traversal, but abstracts out detailed modeling within the switches. The switches are modeled in simple/PerfectSwitch.cc while the links are modeled in simple/Throttle.cc. The flow-control is implemented by monitoring the available buffers and available bandwidth in output links before sending. Garnet is a detailed interconnection network model inside gem5. It consists of a detailed fixed-pipeline model, and an approximate flexible-pipeline model.

Topic 11 & 12

- Topic 11: Analyze the simple network model in gem5 Ruby memory system
- Topic 12: Analyze the garnet network model in gem5 Ruby memory system


ZHEJIANG UNIVERSITY
1897